Computer Science Department

TECHNICAL REPORT

On The Shadow CPU Approximation For Modelling Priority Scheduling In Computer Systems

by

Paul G. Spirakis

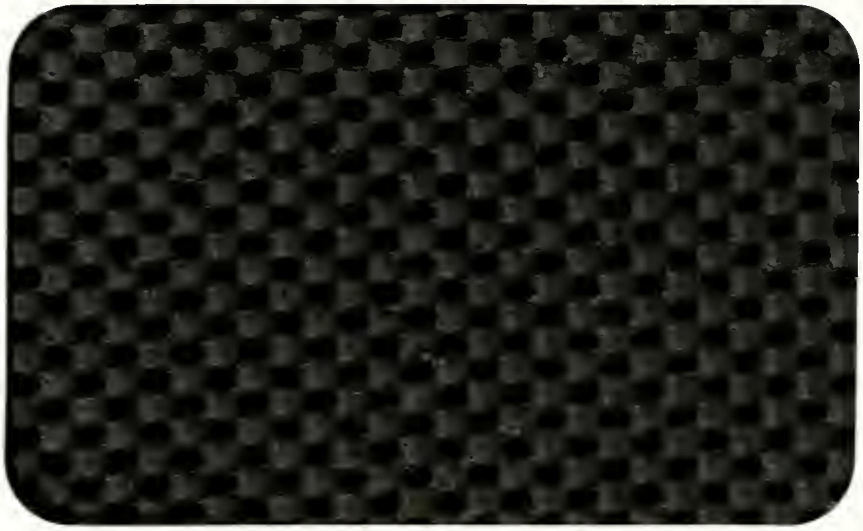
January 1983

Technical Report No. 55

NEW YORK UNIVERSITY

CSD TR-55
C.1

Department of Computer Science
Courant Institute of Mathematical Sciences
251 MERCER STREET, NEW YORK, N.Y. 10012



On The Shadow CPU Approximation
For Modelling Priority Scheduling
In Computer Systems

by

Paul G. Spirakis

January 1983

Technical Report No. 55

This work was supported in part by the NSF Grant NSF-MCS79-21024 and the ONR Contract N00014-80-C-0674.

C.1

ABSTRACT

A sound performance evaluation methodology is essential to the design and use of computer systems. Among performance evaluation methods, queuing network models have been widely recognized as cost and time effective tools, capable of predicting ability and necessary complements of simulations. Algorithms for efficiently estimating the throughput, response time and other measures of performance have been devised for a broad range of queuing network models. Although this range is very rich and adequate for modelling many realistic features of a computer system, nevertheless it has major limitations. An important one is the inability to model pre-emptive priority scheduling disciplines. Although exact closed-form solutions for such network models are open reseach problems, several approximation techniques have been proposed. An important one was Sevcik's [Sevc, 77] "shadow CPU" technique, which admits a product form solution and gives performance bounds which are quite tight in many circumstances. In this paper, we extend the shadow CPU model to an optimal approximation of the priority model, within the restrictions of the product form. We propose a computationally efficient sub-optimal approximation technique which gives tighter performance bounds compared to the original shadow CPU technique. We also investigate the limitations of such shadow server-product form techniques for modelling priorities.

1. Introduction

With the advancement of technology, computer systems have become exceedingly complex and performance evaluation can no longer be done on an informal basis. In addition, detailed simulation of complex systems is very expensive. Fortunately, analytic models based on queuing networks have been shown in several cases to provide predictive ability comparable to that of extensive simulations, yet more efficient and cost-effective ([Giam, 76], [Rose, 76]). In addition, a lot of approximation techniques have been devised to model situations for which closed-form computationally efficient solutions are not known. (See, for example, [Koba, 74], [Gele, 75], [Re, Ko, 74], [Sa,Ch, 75], [Sh,Bu, 77], [Reis, 81], etc.)

In the past few years, the early work on analysis of queuing networks (see [Jack, 63], [Go,Ne, 67], [Klei, 75], [Klei, 76], [Ba,Mu, 73]) has been substantially extended. Buzen ([Buze, 73]) presented a very efficient computational method for deriving performance measures from closed queuing networks with exponential servers and a single customer class ("product form," first analyzed by [Jack, 63]. Basket, Chandy, Muntz and Palacios ([BCMP, 75] demonstrated that certain more general networks have the product form property of state probabilities. Although the class of queuing network models admitting a product form solution is very rich ([Court, 72], [Re,Ko, 75]), nevertheless it has major limitations of which an important one is the inability to accommodate the pre-emptive priority scheduling discipline where the jobs from the higher priority class pre-empt the jobs from the lower class and within each class the scheduling is FCFS (First Come First Serve). Queuing networks with pre-emptive priority scheduling can be analyzed by the method of global balance equations (see [Klei, 75]). This method results in a system of linear equations which can be numerically solved to determine steady state probabilities of the system states. However, the cardinality of the system state space grows exponentially with the number of service centers, customers and classes.

Although priority scheduling is very important and basic in most multi-programming computer systems, no exact closed form solutions of networks with priorities are yet known. Thus, researchers attempted to efficiently model priority scheduling through approximation techniques (see [Mitra, 72], [Reise, 76], [Sa,Ch, 75], [Sevc, 77]). Sevcik's work [Sevc, 77], provided a sequence of approximations using queuing networks with product form state probabilities. Some of those approximations used the idea of introducing a second central processor (the shadow CPU) for the exclusive use of the low priority class.

In this paper we extend the power of the shadow CPU approximation to provide an optimal approximation to the priority model, which is pessimistic for the performance of the high priority class and admits a product form solution. We also provide a suboptimal model which is computationally efficient and outperforms the original (modified) shadow CPU of [Sevc, 77]. We also investigate the limitations of the technique of using a separate server for the lower priority class and staying within the product form class of networks.

2. The Priority Scheduling Model and Sevcik's Work

We shall use a model proposed by [Sevc, 77] that contains very little of the complexity of an actual system but still reflects the intrinsic difficulty of analytically approaching priority scheduling. The queuing network is a closed cyclic network of two servers: a CPU and an I/O server. There are just two customer classes, with class 1 having pre-emptive priority over class 2 at the CPU, while all scheduling is FCFS at the I/O. That is, the central processor will be devoted to class 1 whenever a class 1 job is available for service. Within each class jobs are served in FCFS order. The average service times at the I/O are the same (X_3) for both classes but the average CPU service times (between I/O requests) are different for the two classes (X_i for class i, i = 1,2). Jobs always return to the CPU after being served at the I/O. We assume, for simplicity, that the service time distribution of the I/O is exponential (of rate $\mu_3 = X_3^{-1}$) and the CPU service for class 1 is exponential (rate $\mu_1 = X_1^{-1}$). For class 2 the same holds, when the CPU service time distribution is conditional to the event of absence of class 1 jobs from the CPU.

This model (called the exact model) has the essential features of a computer system with an interactive and a batch class of jobs, where the interactive jobs have pre-emptive priority at the CPU. The I/O server is a crude model of several channel/device groups. Let N_i be the total number of class i customers (i = 1,2) and let N_i be constant (fixed multi-programming level, heavy load assumption). Without loss of generality we will quantify the quality of an approximation to the exact model by the degree it approximates the average response time of the class 1 jobs at the CPU.

[Sevc, 77] proposed that in order to eliminate contention at the CPU due to class 2 jobs (but without ignoring them), a virtual CPU should be introduced (shadow CPU). The shadow CPU exclusively serves the class 2 jobs (with mean rate $\mu_2 = X_2^{-1}$). We call this the shadow CPU (SCM) model. In the SCM model class 2 jobs receive unrealistically good service at their own CPU, thus SCM is a pessimistic model for class 1. Sevcik proposed that in order to get a better approximation, we should increase the mean service time of the class 2 jobs from X_2 to $\frac{X_2}{1 - \rho_1}$, where ρ_1 is the CPU utilization of class 1. By this modification (modified shadow model, MSM) we slow down the CPU performance of class 2, getting a better model for class 1. While ρ_1 is not known a priori, an iterative approach could be used to reach a self-consistent class 1 CPU utilization. MSM differs from the exact model only in that the times between departures of class 2 jobs are exponentially distributed over the busy period of the shadow CPU. As [Sevc, 77] remarks, in the exact model, the class 2 interdeparture times have higher variance (since they are interrupted by class 1 busy periods at the CPU) and (since increasing variance decreases throughput) class 2 progresses faster in the MSM than in the exact model and produces larger I/O queues, causing the MSM to be a pessimistic model for class 1.

3. The Generalization of the MSM Model

To provide a conceptual basis of the various results, we shall generalize the SCM and MSM by introducing a load dependent MSM model (LDSM, load dependent shadow model) in which the mean service rate of the shadow server, when i class 2 customers are in the shadow server, is $\mu_2(i) = \beta(i) \cdot \mu_2$ for $1 \leq i \leq N_2$ with $\mu_2 = X_2^{-1}$ and $\beta(i) > 0$ for $1 \leq i \leq N_2$. The functions $\beta(i)$ (yet to be specified) allow us to get a spectrum of models which includes the pessimistic as well as the optimistic models, within the product-form restriction. (See also [Sh, 79].)

Table 1

(1) all $\beta(i) = 0$ => optimistic model

(2) all $\beta(i) = 1$ => SCM

(3) all $\beta(i) = 1-\rho_1$ => MSM

(4) $0 < \beta(i) < 1-\rho_1$ => product form models bounded by the pessimistic MSM and the optimistic model.

(5) All i, $1-\rho_1 \leq \beta(i) \leq 1$ => All pessimistic models between SCM and MSM.

Notice that the exact results will be best approximated by the category (4).

Lemma 1. The mean service rate of the class 2 jobs in the exact model is $\bar{\mu}_2 = \mu_2(1 - \rho_1)$, where ρ_1 = class 1 CPU utilization.

Proof. $\bar{\mu}_2 = \sum_S p(S)\mu_2(S)$, S = state of the exact model and $\mu_2(S) = 0$ if S implies $n_1 > 0$ where n_1 is the number of class 1 customers at the CPU and $\mu_2(S) = \mu_2$ if S implies $n_1 = 0$. Hence $\bar{\mu}_2 = \sum_{S \in K} \mu_2 \cdot p(S)$ where $K = \{S: n_1 = 0\}$, i.e.,

$$\bar{\mu}_2 = \mu_2 \cdot \text{Prob}\ (n_1 = 0) = \mu_2(1 - \rho_1)$$

Let us denote by server 1, 2 or 3 the CPU, shadow CPU or I/O respectively. Let n_j be the number of customers at server j ($j = 1,2,3$) at state S. Let $p_k(i)$ be the marginal probability of i customers at server k. The mean rate of the shadow CPU of any LDSM model is

$$\mu_2^* = \sum_{i=1}^{N_2} p_2(i)\beta(i)\mu_2 .$$

Definition. An LDSM is called reasonable if its μ_2^* satisfies the restriction:

(R1) $$\mu_2^* = \mu_2(1 - \rho_1)$$

or, in other terms

$$\sum_{i=1}^{N_2} p_2(i)\beta(i) = p_1(0) = 1 - \rho_1$$

For example, MSM is a reasonable LDSM whereas SCM is not. Each choice of $B = \{\beta(i),\ i = 1,2,\dots,N_2\}$ in Table 1, category (4), corresponds to a product form LDSM whose class 2 customers still depart from the shadow server independently of the population count of the "real" CPU. In fact, each LDSM has an "illegal" (nonzero) mean transition rate of class 2 customer departures out of states S for which $n_1 > 0$. Let us denote the probability of a state S by p(S,B) to indicate that p is a function of the set B of load dependent factors of the shadow. The set of feasible system states, Σ, is a partition of two disjoint sets Σ_+ and Σ_o where

$$\Sigma_+ = \{S \in \Sigma \text{ with } n_1 > 0\}, \qquad \Sigma_o = \{S \in \Sigma \text{ with } n_1 = 0\} .$$

Let $\Sigma_+^i = \{S \in \Sigma_+ \text{ with } n_2 = i\}$, $\Sigma_o^i = \{S \in \Sigma_o \text{ with } n_2 = i\}$ for $i = 0,1,2,\dots,N2$, and $\Sigma^i = \Sigma_+^i \cup \Sigma_o^i$. Then each choice of B implies an illegal mean rate

$$\text{(EQ1)} \qquad ir = \sum_{i=1}^{N_2} \beta(i)\Big(\sum_{S \in \Sigma_+^i} p(S,B)\Big)$$

For any set B of positive $\beta(i)$'s, ir is positive, hence the solution of the exact model cannot be obtained by any choice of $\beta(i)$'s in Table 1. However, the exact solution can be approximated in an optimal way if we select B to minimize ir, subject to the constraint of reasonable LDSM's. Notice that the constraint can be written as

$$\text{(R2)} \qquad \sum_{i=1}^{N_2} \beta(i) \Big(\sum_{S \in \Sigma^i} p(S,B) \Big) = \sum_{S \in \Sigma_0} p(S,B)$$

i.e.,

$$ir + \sum_{i=1}^{N_2} \beta(i) \Big(\sum_{S \in \Sigma_0^i} p(S,B) \Big) = \sum_{i=0}^{N_2} \sum_{S \in \Sigma_0^i} p(S,B)$$

or

$$\text{(EQ2)} \qquad ir = \sum_{i=1}^{N_2} (1 - \beta(i)) \cdot \Big(\sum_{S \in \Sigma_0^i} p(S,B) \Big) + \sum_{S \in \Sigma_0^0} p(S,B)$$

Corollary 1. An optimal (with respect to ir) reasonable LDSM approximation to the exact priority model can be obtained by solving the following minimizatin problem:

Choose $B = \{\beta(i),\ i = 1,\ldots,N2\}$ with $\beta(i) > 0$ to minimize ir (given by (EQ1)) under the restriction (R2), or to minimize ir (given by (EQ1)), under the constraint (EQ1) = (EQ2).

[BCMP, 75] proved that the probability of an aggregate system state $S = (n_1,n_2,n_3)$ is of the form

$$p(S,B) = C \cdot g_1(n_1) \cdot g_2(n_2,B) \cdot g_3(n_3)$$

where $g_i(n_i)$ depends only on the conditions at server i. In any LDSM all servers are of type 1 hence (e.g., for the shadow),

$$g_2(n_2,\beta) = e_2^{n_2} \frac{1}{\mu_2^{n_2}} \cdot \frac{1}{\prod_{k=1}^{n_2} \beta(k)}$$

where e_2 is the relative visit count at the shadow. (It could be taken equal to μ_2). We conclude that the form of the function to minimize is (within a constant)

$$\frac{A_1}{\beta(1)} + \frac{A_2}{\beta(1)\beta(2)} + \ldots + \frac{A_{N_2-1}}{\beta(1)\beta(2)\ldots\beta(N_2-1)}$$

where $A_1, A_2, \ldots, A_{N_2-1}$ depend on N_1, N_2, μ_1, μ_2 and μ_3. It is easy to prove that

Remark: There is a positive $\gamma = \gamma(N_1,N_2,\mu_1,\mu_2,\mu_3)$ (which does not depend on a particular state S) such that $ir \leq \gamma\, ir'$, where

$$ir' = \frac{1}{\beta(1)} + \frac{1}{\beta(1)\beta(2)} + \ldots + \frac{1}{\beta(1)\ldots\beta(N_2)}$$

(To prove it, consider the form of (EQ1) and replace numerators (which are < 1) by ones.) We shall provide a suboptimal solution, which is computationally easy, by minimizing ir' instead of ir, subject to R1.

Theorem 1. The set $B = \{\beta(i),\ i = 1,\dots,N_2\}$ with

$$\beta(i) = \frac{2(N_2 + 1 - i)}{N_2(N_2 + 1)} \cdot \frac{1}{p_2(i)} \cdot (1 - \rho_1)$$

minimizes π' with respect to R1.

Proof: See Appendix 1. Note that Theorem 1 gives the $\beta(i)$ as functions of the unknown quantities $p_2(i)$ and ρ_1. An iteration process is needed to get self-consistent $\beta(i)$'s (of the same form as in [Sevc, 77].

Several numerical examples showed that the (reasonable) LDSM with B as in Theorem 1 is a pessimistic approximation to the exact solution, better than the MSM under most load conditions, and of the same computational effort. To find the optimal approximation one can use Lagrange multipliers and Newton-type numerical solution of the obtained equations. This is still computationally more efficient than the global balance method, since the number of equations is proportional to N_2 and only a few iterations are usually needed.

4. Limitations of the Product Form Shadow CPU Type Models

By [BCMP, 75] the aggregate state probability distribution remains the same even if the shadow CPU is considered to be a type 2 server (instead of type 1), since the state descriptions are equivalent. One then may hope that, by choosing a suitable service time distribution for the shadow CPU (with rational Laplace transform) we would be able to get the exact solution. This is unfortunately not so, since (for type 2 servers) any distributions (with rational transforms) give the same results as the exponential service time distribution. This implies that the best we can achieve by a type 2 shadow is our optimal approximation model of Section 3. It is impossible to get the exact solution since no exponential service time distribution zeros in. However, one could approximate the exact solution by the following iterative scheme (which uses PS instead of FCFS for the shadow):

Process STAGE

initialization: Let the shadow CPU be as in the MSM model.

Let $i = 1$.

1. Find the $1^{st}, 2^{nd}, \ldots, i + 1^{th}$ moments of the idle periods of the class 1 server (server 1) based on the solution of the previous model. Relate this to the $1^{st}, 2^{nd}, \ldots, i + 1^{th}$ moments of the service time of the shadow CPU by inverting and multiplying by the corresponding moment of an exponential variable of mean X_2.

2. Add one more stage (exponential) to the shadow CPU, in such a way to obtain the moments estimated above. (By using, e.q., Cox's stages representation).

3. Go to step 1 after setting i = i + 1, unless you have agreement with the "exact" model up to a desirably high moment.

The networks obtained by STAGE are product form (of a type 2 shadow). Hence, by our previus remarks, they can never be out of the class of solutions obtained by our LDSM approximations. This implies that not only the service time distribution of class 2 jobs at the exact model does not have a rational Laplace transform, but also that its higher moments are important in determining the distribution. This provides evidence that the exact solution is not a product form solution, as our next theorem proves:

Theorem 2: The exact priority model does not admit a product form solution, even if no assumptions are made about the service time distributions of the CPU and I/O, besides the pre-emptive priority of class 1 over class 2.

Proof. We shall prove that the exact model does not satisfy "Exponential Local Balance" (ELB). The rest follows by a theorem of [Ch,Sa, 78] that product form implies ELB. We state the ELB property:

A queuing discipline is ELB iff

$$R(k/s)R(j/_{s-k}) = R(j/s)R(k/_{s-j})$$

for all s, s-j, s-k states in the state space and all classes k and j, where $R(k/s)$ is the total rate of service for class k customers, given s, and s-k is the state with one less class k customer.

Indeed, consider the state s^* with one class 1 customer at the CPU and an I/O queue with a class 1 customer first and a class 2 customer second. This state is always reachable from any initial state of the priority model provided both $N_1,N_2 > 2$. We have $R(2/s^*) = 0$ for the total rate of class 2 customers. But $R(1/s^*) > 0$ and $R(2/s^* - 1) > 0$. Hence for $s = s^*$ and $k = 2$, $j = 1$ the ELB condition is violated, so the exact model is not product form.

REFERENCES

[Ba,Mu, 73] Baskett, F. and Muntz, R.R., "Networks of queues," Proc. 7th Annual Princeton Conference on Information Sciences and Systems, Princeton Univ., Princeton, N.J., March 1973, pp. 428-434.

[BCMP, 75] Baskett, F., Chandy, K.M., Muntz, R.R. and Palacios, F.G., "Open, closed and mixed networks of queues with different classes of customers," J. ACM, Vol. 22, No. 2, April 1975, pp. 248-260.

[Buze, 73] Buzen, J.P., "Computational algorithms for closed queuing networks with exponential servers," J. ACM, Vol. 16, No. 9, Sept. 1973, pp. 527-531.

[Re,Ko, 75] Reiser, M. and Kobayashi, H., "Queuing networks with multiple closed chains: Theory and computational algorithms," IBM J. of Res. and Develop., Vol. 19, No. 3, May 1975, pp. 283-294.

[Court, 72] Courtois, P.J., "on the near-complete decomposability of networks of queues and of stochastic models of multi-programming computer systems," Ph.D. Thesis, Dept. of Comp. Sci., Carnegie-Mellon University, Pittsburg, Pa., Nov. 1972.

[Gele, 75] Gelenbe, E., "On approximate computer system models," J. ACM, Vol. 22, No. 2, 1975, pp. 261-263.

[Giam, 76] Giammo, T.P., "Validation of a computer performance model of the exponential queuing network family," Proc. Int. Symp. on Computer Performance, Modeling, Measurement and Evaluation, March 1976, pp. 44-58.

[Go,Ne, 67] Gordon, W.J. and Newell, G.F., "Closed queuing systems with exponential servers," *Oper. Res. 15*, 2, April 1967, pp. 254-265.

[Jack, 63] Jackson, J.R., "Job shop-like queuing systems," *Mgmt. Sci. 10*, 1, Oct. 1963, pp. 131-142.

[Klei, 75] Kleinrock, L., "Queuing systems," *Theory*, Vol. I, Wiley Interscience, N.Y., 1975.

[Klei, 76] Kleinrock, L., "Queuing systems," *Computer Applications*, Vol. II, Wiley Interscience, N.Y., 1976.

[Koba, 74] Kobayashi, H., "Application of diffusion approximation to queuing networks, Part I: Equilibrium queue distributions," *J. ACM*, Vol. 21, No. 2, April 1974, pp. 316-328.

[Mitra, 72] Mitrani, I., "A queuing model of priority multi-programming," TR-41, Univ. of Newcastle Upon Tyne, Dec. 1972.

[Re,Ko, 74] Reiser, M. and Kobayashi, H., "Accuracy of the diffusion approximation for some queuing systems," *IBM J. Res.*

and Develop., Vol. 18, No. 2, March 1974, pp. 110-124.

[Reise, 76] Reiser, M., "Modelling of computer systems with QNET4," IBM Res. Report RC 5081, 1976.

[Reise, 81] Reiser, M., "Mean value analysis and convolution method for queue-dependent servers in closed queuing networks," North Holland Co. Performance Evaluation 1 (1981), pp. 7-18.

[Sa, Ch, 75] Sauer, C.H. and Chandy, K.M., "Approximate analysis of central server models," IBM J. Res. and Develop., Vol. 19, No. 3, May 1975, pp. 301-313.

[Ch, Sa, 78] Chandy, K.M. and Sauer, C.H., "Approximate methods for analyzing queuing network models of computing systems," Comp. Surveys, Vol. 10, No. 3, Sept. 1973, pp. 281-317.

[Sevc, 77] Sevcik, K., "Priority scheduling disciplines in queuing network models of computer systems," Info. Proc. 77 IFIP, North-Holland Publ. Co., 1977.

[Sh, 79] Shum, A.W., personal communication.

[Sh, Bu, 77] Shum, A.W. and Buzen, J.P, "The EPF technique: A method for obtaining approximate solutions to closed queuing networks with general service times," Proc. Int. Symp. Computer Performance, Modelling, Measurement and Evaluation, 1977, North-Holland Publ. Co., Amsterdam, The Netherlands, pp. 201-221.

[Rose, 76] Rose, C.A., "Validation of a queuing model with classes of customers," Proc. Int. Symp. on Computer Performance, Modelling, Measurement and Evaluation, March 1976, pp. 318-326.

APPENDIX 1

Proof of Theorem 1:

The constraint becomes

$$\sum_{i=1}^{N_2} \beta(i) \left(\frac{p_2(i)}{1-\rho_1}\right) = 1$$

By the fact that arithmetic averages of positive numbers are greater than (or equal to) geometric, we get

$$\text{(IN1)} \quad \frac{ir'}{N_2} > \left[\beta(1)\cdot(\beta(1)\beta(2))\cdot \; \ldots \; \cdot(\beta(1)\ldots\beta(N_2))\right]^{-\frac{1}{N_2}}$$

Hence ir' gets its minimum when (IN1) becomes an equality, i.e., when

$$\beta(1)^{N_2}\,\beta(2)^{N_2-1}\,..\,\beta(N_2) \text{ maximizes.}$$

From the form of the constraint, this happens when

$$\frac{\beta(i)}{N_2+1-i} \cdot \frac{p_2(i)}{1-\rho_1} = \frac{1}{\sum_{i=1}^{N_2} (N_2+1-i)} \quad \text{for all } i,$$

$$\Rightarrow \beta(i) = \frac{2(N_2+1-i)}{N_2(N_2+1)} \cdot \frac{1-\rho_1}{p_2(i)} .$$

Note: Given $a_1x_1 + a_2x_2 + \ldots + a_Nx_N = 1$ (and $a_i, x_i > 0$) the product

$x_1^{\lambda_1} \quad x_2^{\lambda_2} \quad \ldots \quad x_N^{\lambda_N}$, maximizes

when $\frac{a_i x_i}{\lambda_i}$ = same for all i's = $\frac{1}{\sum_{i=1}^{N} \lambda_i}$.

This book may be kept
NOV 1 1983
FOURTEEN DAYS
A fine will be charged for each day the book is kept overtime.
GAYLORD 142
PRINTED IN U.S.A.

NYU c.1
Comp. Sci. Dept.
TR-55 Spirakis
On the shadoe CPU approximation for modelling...

NYU c.1
Comp. Sci. Dept.
TR-55 Spirakis

AUTHOR
On the shadow CPU approxi-

TITLE
mation for modelling ...

DATE DUE	BORROWER'S NAME

N.Y.U. Courant Institute of
Mathematical Sciences
251 Mercer St.
New York, N. Y. 10012